

REMARKS

Claims 1-15 are all the claims pending in the application. By this Amendment, Applicant editorially amends claims 5 and 14 to overcome the rejection under 35 U.S.C. § 112, second paragraph.

Summary of the Office Action

The Examiner withdrew all of the previous rejections. However, the Examiner found new grounds for rejecting claims 1-15. In particular, claims 5, 14 and 15 are rejected under 35 U.S.C. § 112, second paragraph and claims 1-15 stand rejected under 35 U.S.C. § 102(b), as being anticipated by USP 5,918,225 to White et al. (hereinafter “White”).

Claim Rejections under 35 U.S.C. § 112

The Examiner rejected claims 5, 14 and 15 under 35 U.S.C. § 112, second paragraph. In particular, the Examiner thought the terms “product” and “combination” to be indefinite. Claims 5 and 14 has been revised, and no longer include the potential indefiniteness mentioned by the Examiner. Therefore, Applicant respectfully requests the Examiner to withdraw this rejection of claims 5, 14 and 15.

Claim Rejections under 35 U.S.C. § 102

The Examiner rejected claims 1-15 under 35 U.S.C. § 102(b) as being anticipated by White. The Examiner’s careful reconsideration is submitted to be appropriate in view of the following comments traversing the rejection. Of these claims, only claims 1, 8 and 14 are independent.

To be an “anticipation” rejection under 35 U.S.C. § 102, the reference must teach every element and recitation of the Applicant’s claims. Rejections under 35 U.S.C. § 102 are proper only when the claimed subject matter is identically disclosed or described in the prior art. Thus, the reference must clearly and unequivocally disclose every element and recitation of the claimed invention.

Claims 1-7

Claim 1 recites a novel combination of elements not found in the cited references. For example, claim 1 recites a *tabulation stage in which, for each data record, a cell of a result array is determined based on the numerical identifiers for that record, and a resulting value stored in the result array cell is incremented*. The Examiner asserts that White teaches a method of data tabulation, as set forth in claim 1. In particular, the Examiner asserts that White's method of inserting the appropriate identifier into the row-oriented cell array and incrementing a reference count (integer) in the lookup value entry (which is being referenced) is equivalent to the tabulation stage, as set forth in claim 1. White's discussion of the indexing method was carefully studied and it is respectfully submitted that this ground of rejection is respectfully submitted to be technically inaccurate.

An illustrative, non-limiting embodiment of the present invention discloses the tabulation unit formulating a resulting array based on a request for tabulation from a user. In particular, the exemplary method teaches initializing the resulting array to zeros. Then using the unique identifiers, an index for the resulting array is computed. An index is a cell identifier (e.g. array[0]), identifying to which cell a certain item belongs. Based on that index, the values in the cells of the resulting array are incremented. Thus, a cell of the resulting array may comprise of a count for intersection of two or more events or data type fields. For example, in the resulting array, one of the cells may hold a number of males in the area A. In short, in the exemplary tabulation stage, values are extracted from the stored format, analyzed and are presented to the user. This passage is provided by way of an example only and is not intended to limit the scope of the claims in any way.

White teaches an improved method of storing data vertically, by columns, instead of horizontally. Each column has a plurality of cells (records). White also teaches having a look up table for tracking unique values in the cells (col. 3, lines 21 to 47; col. 4, lines 8 to 10). That is, White teaches storing the unique value in a look-up table and assigning a small unique integer identifier for that unique value. The look up table will contain the relationships between the unique value and the integer identifier as well as a count identifying a number of occurrences of

this particular unique value and the actual data record will hold the integer identifier. When the user requests to retrieve value or values, the small integer value is used to index the look up table to locate the unique value corresponding to this small integer identifier (col. 6, lines 60 to 64).

In short, White teaches storing data more effectively by storing data in columns with short integer identifiers and having a look-up table correlating integer identifiers and the unique value originally present in a data cell. Moreover, the look-up table stores a count, which is a number of occurrences of the unique value in data records.

However, with respect to retrieving data, White only teaches as recited in the section g, titled “use during query processing”:

[w]hen the index is being used for queries, the system will pass in a row number. The database system will, in turn, determine which chunk of the row-ordered table contains that row and bring the chunk into memory. Using the assigned small integer found at the row location, the system can readily determine the cell value by indexing into the value lookup table (col. 52, lines 41 to 48).

In other words, White only teaches a conventional query processing, using the small integer to access the look-up table to obtain data cell unique value. White clearly fails to disclose having a resulting array. Moreover, White fails to disclose anything about the cell of the resulting value being incremented. In short, White is not related to use during query processing, instead White focuses on a more efficient way of storing data.

The Examiner alleges that step 215 of the White’s indexing method is equivalent to the tabulation process, as set forth in claim 1. However, with respect to step 215, White discloses:

[a]t step 615, the method inserts the appropriate identifier into the row-oriented cell array; a reference count (integer) stored at the lookup value entry (which is being referenced) is incremented. By using reference counts, the system can perform

clean-up of entries in the value lookup table which are no longer being referenced. Since the column(s) of the input table can be reconstituted from the cell array, the underlying column(s) can be (optionally) deleted, in effect compressing the input table (col. 53, lines 36 to 45).

In other words, White teaches replacing unique values in the data table with short integers for a more efficient storage. In addition, White teaches creating a mapping file, a reference table, which will store these unique values and correlate them with this short integer. Also, the look up table stores a reference count for clean-up of entries (deleting values in the look-up table, which are no longer being references). In short, White teaches a mapping file and not a resulting array because the count is determined based on the unique values, which are then replaced with integer identifiers and at the same time the counter is incremented based on this encountered unique value.

Therefore, *“a cell of a result array is determined based on the numerical identifiers for that record, and a resulting value stored in the result array cell is incremented”*, as set forth in claim 1 is not disclosed by White, which lacks a resulting array being determined based on the unique identifiers. The look up table in White is determined based on the original unique values in the data records. For at least these exemplary reasons, independent claim 1 is patentably distinguishable from White, and it is appropriate and necessary for the Examiner thus to withdraw this rejection of independent claim 1. Also, claims 2-7 and 13 are allowable at least by virtue of their dependency on claim 1.

In addition, with respect to the dependent claim 7, the Examiner alleges that Fig. 6B and col. 53, lines 36 to 40 and col. 54, lines 50 to 53 of White disclose the formula, as set forth in the dependent claim 7 (see page 4 of the Office Action). Col. 53, lines 36 to 40, were cited and explained above. Col. 54, lines 50 to 53 recite “wherein each unique entry of the value lookup table further stores a reference count for indicating how many times the entry’s unique value occurs in the particular column being indexed”, emphasis added. These lines deal with creating

a reference table, which stores a correlation between unique value and the assigned short integer, and a count value (indicating number of occurrences of this unique value).

However, not only does White fail to disclose a resulting array but also White clearly fails to teach a cell identity being equal to $K1+D1K2+D1D2K3+\dots+D1D2\dots Dn-1Kn$, where $K1\dots Kn$ are numerical identifiers for the record in the selected fields and $D1\dots Dn$ are number of distinct values in the selected fields. White does not teach anything about a cell identity. It appears that in White, just like in the conventional mapping files, the next empty or available slot in the look up table is used. No calculations as to the cell identity is disclosed in White. In addition, White's look up table does not have a count for the number of distinct values of a particular field. White only teaches counting a number of occurrences of a particular unique value and correlating unique values with short integers. White's look up table does not store a number of distinct values for a particular field and there is no disclosure in White that a count for a number of distinct values for a selected field is calculated.

In short, White is directed to reducing the storage size and thus only teaches replacing a unique value (e.g. state Alabama) with a unique integer (e.g. 0), (col. 49, lines 3 to 14). White fails to clearly and unequivocally disclose calculating the cell identity, as set forth in claim 7. For at least this additional reason, it is appropriate and necessary for the Examiner to withdraw this rejection of claim 7.

Claims 8-13

This rejection is now traversed with respect to independent claim 8. Claim 8 recites: *a tabulation processor in which, for each data record, a cell of a result array is determined based on the numerical identifiers for that record, and a resulting value stored in the result array cell is incremented.* This limitation is similar to the limitation of a tabulation stage in which a cell of a result array is determined based on the numerical identifiers for that record, and incrementing a resulting value stored in the result array cell is incremented, recited in claim 1. Since claim 8 contains features that are similar to the features argued above with respect to claim 1, those arguments are respectfully submitted to apply with equal force here. For at least substantially the

same exemplary reasons, therefore, the Examiner is respectfully requested to reconsider and to withdraw this rejection of independent claim 8 and its dependent claims 9-12.

Claims 14-15

Next, this rejection is traversed with respect to independent claim 14. Claim 14 was amended to ameliorate the Examiner's concerns that the term "combination" is indefinite. Claim 14, as now amended, recites: *said combination is an intersection of at least one said unique identifier of one individual data type field with said unique identifier of another individual data type field*. White only teaches a look up table, which correlates a unique value with unique identifier. White fails to disclose a resulting array with at least one cell being allocated for a combination, as forth in independent claim 14. For at least this exemplary reason, therefore, it is appropriate and necessary for the Examiner to reconsider and to withdraw this rejection of independent claim 14 and its dependent claim 15.

In addition, with respect to the dependent claim 15, the Examiner alleges that White's Fig. 6A, items 601 and 602 are equivalent to a mapping file (which stores relationships between each of the distinct data values and numerical identifiers) for each of said data type field, as set forth in claim 15. White's discussion of establishing a default size for the identifier and allocating memory for the look up table has been carefully studied, and such teachings are very dissimilar.

With respect to items 601 and 602 of Fig. 6A, White discloses at col. 52, lines 50 to 63:

At the outset, the method establishes a default size (data width) for the uniqueness identifier, as shown at step 601. In the currently-preferred embodiment, the data width is initially set equal to 8 bits (i.e., 1 byte). In a corresponding manner, the method allocates appropriate resources (e.g., memory allocation) for maintaining a value lookup table. For an identifier size of 8 bits, the value lookup table should accommodate no more than 255 entries (or 256 if 0 is not reserved). Each entry within the

table is constructed to store a data value from the underlying user table (e.g., alphanumeric text string) together with a reference or use count (e.g., integer value), emphasis added.

In other words, White discloses determining a default size for the unique identifier at step 601 and allocating appropriate resources (memory allocation) at step 602. However, these steps of White do not disclose storing a relationship between said distinct value and said corresponding numerical identifier.

Moreover, the Examiner alleged that the look-up table of White is similar to the resulting array, as set forth in independent claim 14 (see page 3 of the Office Action). Now, the Examiner alleges that this look up table corresponds to a mapping file. It is respectfully submitted that White's look up table cannot be both a resulting array and a mapping file at least because of claim differentiation and because the mapping file (which holds unique identifiers) is used to create a resulting array. For at least this additional exemplary reason, it is appropriate and necessary for the Examiner to withdraw this rejection of claim 15.

Conclusion and request for telephone interview


In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a personal or telephone interview, the Examiner is kindly invited to contact the undersigned attorney at the telephone number listed below.

Amendment Under 37 C.F.R. § 1.116
U.S. Application No.: 09/582,716

Attorney Docket No.: Q58912

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,



Kelly G. Hyndman
Registration No. 39,234

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE

23373

CUSTOMER NUMBER

Date: June 15, 2004